ID721001: Mobile Application Development

Project: Travelling Application Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	Application contains comprehensive & robust evidence on the following: • Opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. • Country data fetched from a GitHub Gist & displayed in a list. • Text translation & text to speech support. • Selection of well-known phrases. • Register a new user, login a user and logout a user. • Google map displaying tourist attractions as markers. • Light & dark mode. • Splash screen with Lottie animation. • Like & favourite a country. • Adaptive launcher icon. • Visually attractive UI. • Published to & downloadable from Google Play Store. • Incorrectly formatted input fields handled. • Appropriate feedback given to a user.	Application contains clear & detailed evidence of functionality on the following: • Opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. • Country data fetched from a GitHub Gist & displayed in a list. • Text translation & text to speech support. • Selection of well-known phrases. • Register a new user, login a user and logout a user. • Google map displaying tourist attractions as markers. • Light & dark mode. • Splash screen with Lottie animation. • Like & favourite a country. • Adaptive launcher icon. • Visually attractive UI. • Published to & downloadable from Google Play Store. • Incorrectly formatted input fields handled. • Appropriate feedback given to a user.	Application contains evidence on the following: • Opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. • Country data fetched from a GitHub Gist & displayed in a list. • Text translation & text to speech support. • Selection of well-known phrases. • Register a new user, login a user and logout a user. • Google map displaying tourist attractions as markers. • Light & dark mode. • Splash screen with Lottie animation. • Like & favourite a country. • Adaptive launcher icon. • Visually attractive UI. • Published to & downloadable from Google Play Store. • Incorrectly formatted input fields handled. • Appropriate feedback given to a user.	Application does not, or does not fully contain evidence on the following: • Opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. • Country data fetched from a GitHub Gist & displayed in a list. • Text translation & text to speech support. • Selection of well-known phrases. • Register a new user, login a user and logout a user. • Google map displaying tourist attractions as markers. • Light & dark mode. • Splash screen with Lottie animation. • Like & favourite a country. • Adaptive launcher icon. • Visually attractive UI. • Published to & downloadable from Google Play Store. • Incorrectly formatted input fields handled. • Appropriate feedback given to a user.

Code Elegance	Kotlin & XML files thoroughly contain no magic numbers/strings & are stored in their appropriate XML files. Application code thoroughly demonstrates code elegance on the following: • Idiomatic use of control flow, data structures & other in-built functions. • Sufficient modularity, i.e., code adheres to DRY, KISS & MVVM. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • Efficient algorithmic approach. • Code comments documented using KDoc. • API keys stored & retrieved from local.properties. • Code formatted Kotlin & XML files. • No dead or unused code.	Kotlin & XML files mostly contain no magic numbers/strings & are stored in their appropriate XML files. Application code clearly demonstrates code elegance on the following: • Idiomatic use of control flow, data structures & other in-built functions. • Sufficient modularity, i.e., code adheres to DRY, KISS & MVVM. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • Efficient algorithmic approach. • Code comments documented using KDoc. • API keys stored & retrieved from local.properties. • Code formatted Kotlin & XML files. • No dead or unused code.	Kotlin & XML files contain some magic numbers/strings & are stored in their appropriate XML files. Application code demonstrates code elegance on the following: • Idiomatic use of control flow, data structures & other in-built functions. • Sufficient modularity, i.e., code adheres to DRY, KISS & MVVM. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • Efficient algorithmic approach. • Code comments documented using KDoc. • API keys stored & retrieved from local.properties. • Code formatted Kotlin & XML files. • No dead or unused code.	Kotlin & XML files contain frequent magic numbers/strings & are not or are not fully stored in their appropriate XML files. Application code does not or does not fully demonstrate code elegance on the following: • Idiomatic use of control flow, data structures & other in-built functions. • Sufficient modularity, i.e., code adheres to DRY, KISS & MVVM. • Adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. • Efficient algorithmic approach. • Code comments documented using KDoc. • API keys stored & retrieved from local.properties. • Code formatted Kotlin & XML files. • No dead or unused code.
Documentation & Git	README file contains comprehensive evidence of: • URL to application's privacy policy. • URL to commented code generated to Markdown using Dokka. • URL to application on Google Play Store. Git commit messages comprehensively formatted & reflect the feature changes in concise detail.	README file contains clear evidence of: • URL to application's privacy policy. • URL to commented code generated to Markdown using Dokka. • URL to application on Google Play Store. Git commit messages clearly formatted & reflect the feature changes in substantial detail.	README file contains evidence of: • URL to application's privacy policy. • URL to commented code generated to Markdown using Dokka. • URL to application on Google Play Store. Git commit messages formatted & reflect the feature changes in detail.	README file does not or does not fully contain evidence of: • URL to application's privacy policy. • URL to commented code generated to Markdown using Dokka. • URL to application on Google Play Store. Git commit messages do not or do not fully formatted & reflect the feature changes.

ID721001: Mobile Application Development

Project: Travelling Application Assessment Marking Cover Sheet

Name:

Date:

Learner ID:

Assessor's Name:

Assessor's Signature:

Criteria	Out Of	Weighting	Final Result
Functionality	10	40	
Code Elegance	10	45	
Documentation & Git/GitHub Usage	10	15	
Final Result			/100
This assessment is worth 60% of the final mark for the Mobile Application Development course.			

Feedback:

Functionality:

Code Elegance:

Documentation & Git Usage: